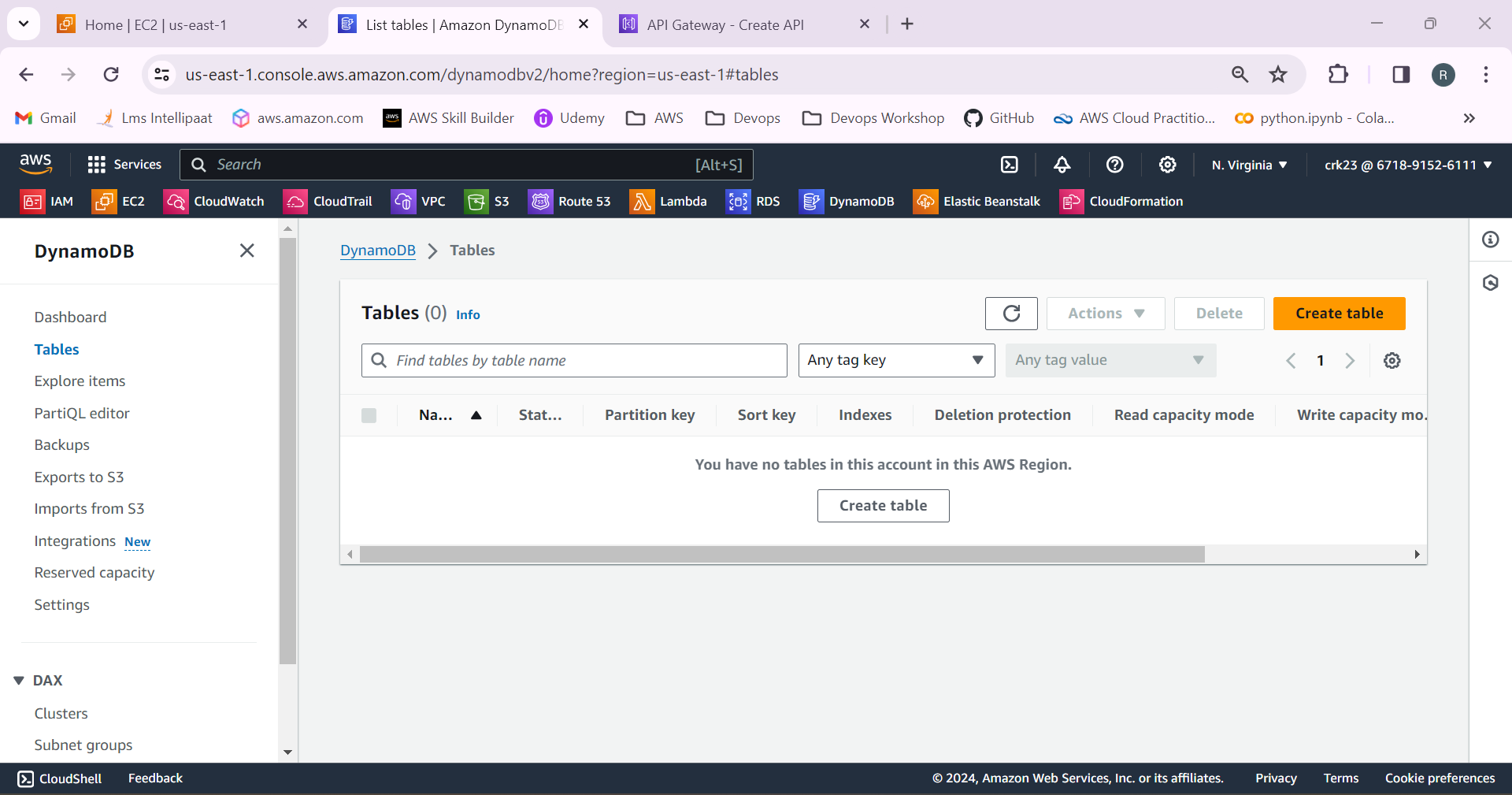
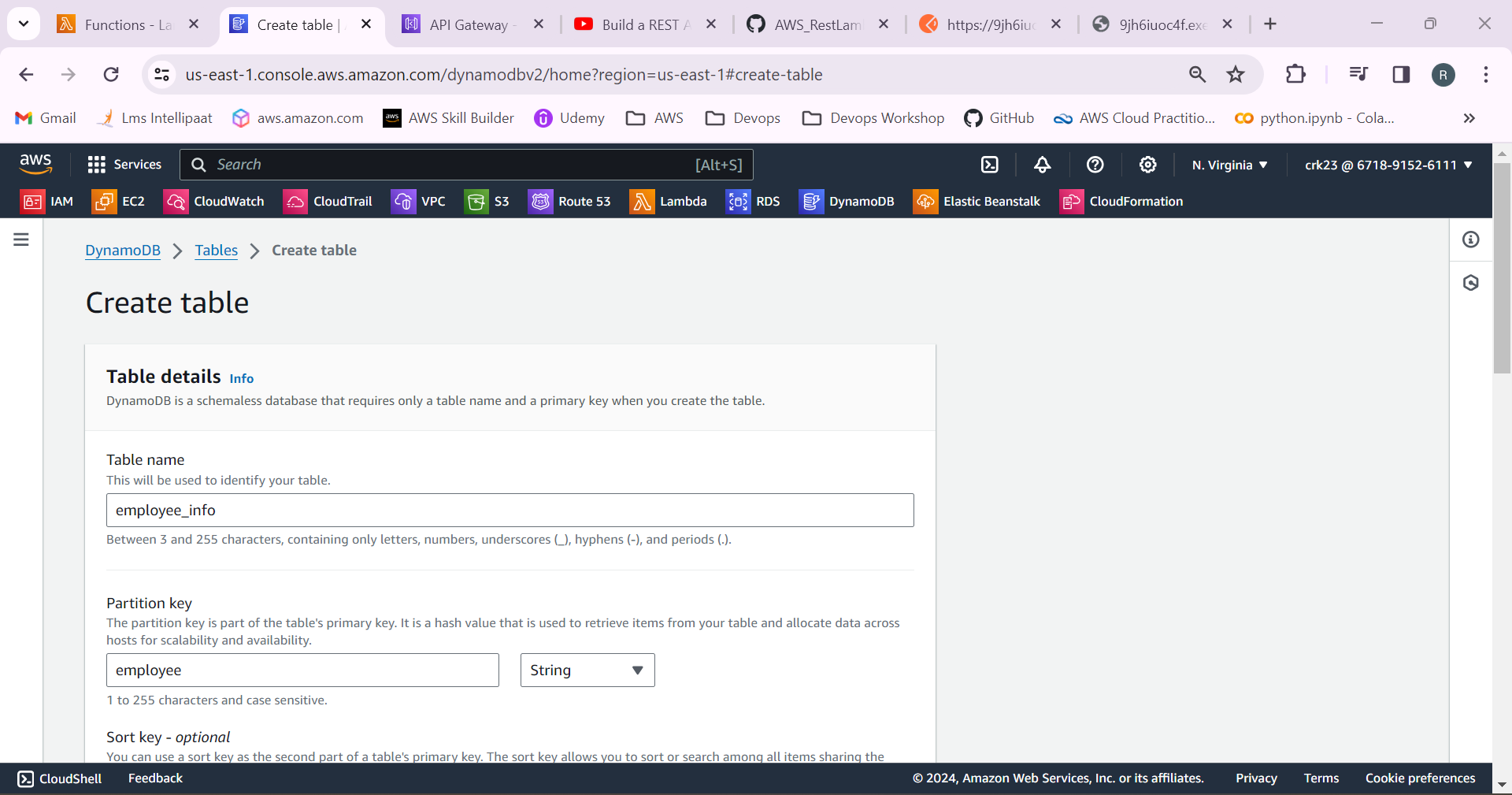
**BUILDING A SERVERLESS API WITH AWS LAMBDA AND API GATEWAY**

**Overview:**

Serverless architecture by creating a RESTful API using AWS lambda for compute and API gateway for endpoint management. Implementing CRUD (Create, Read, Update, Delete) operations on DynamoDB.

1.Create a DynamoDB table with partition key as employee.





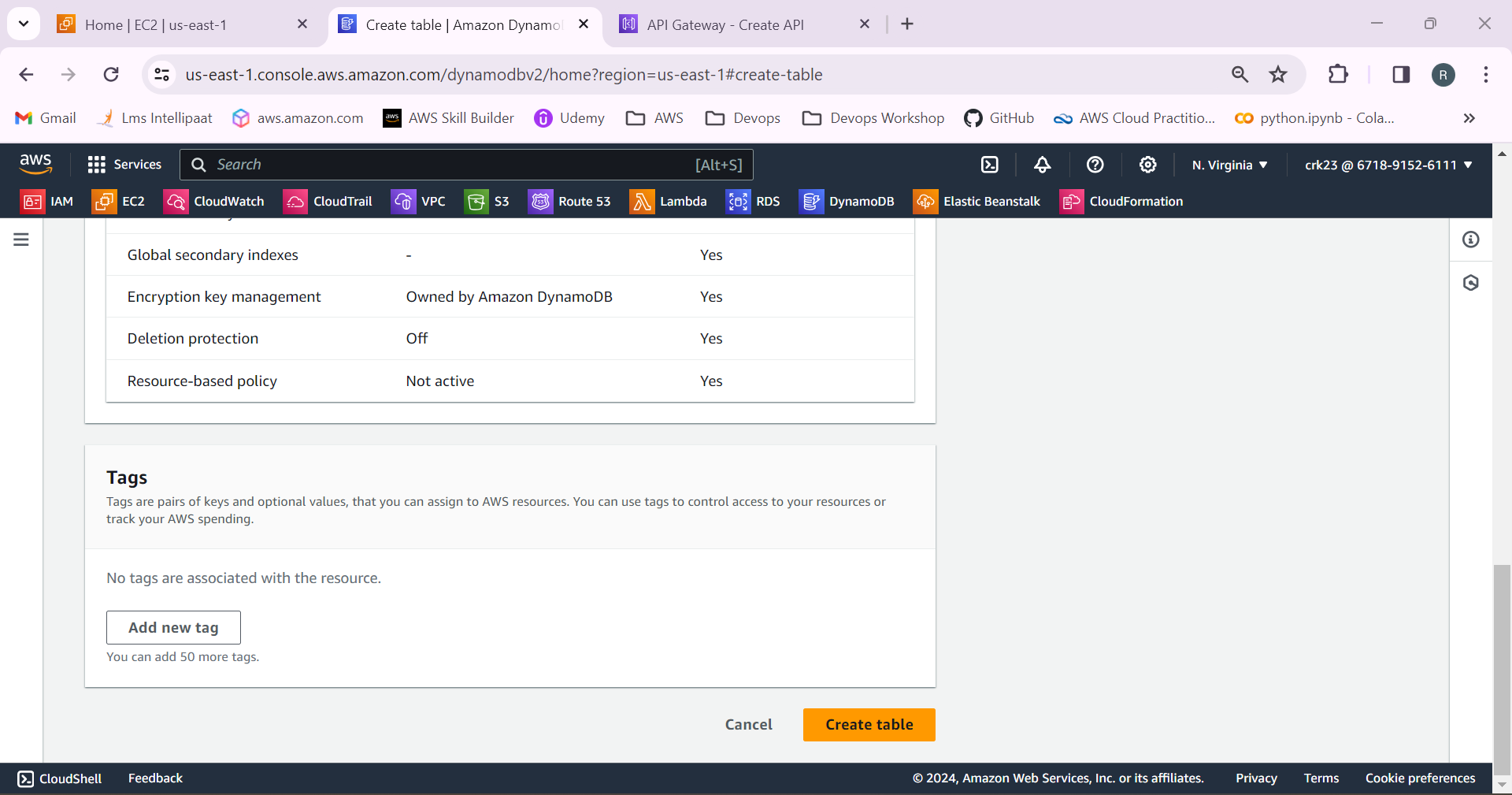
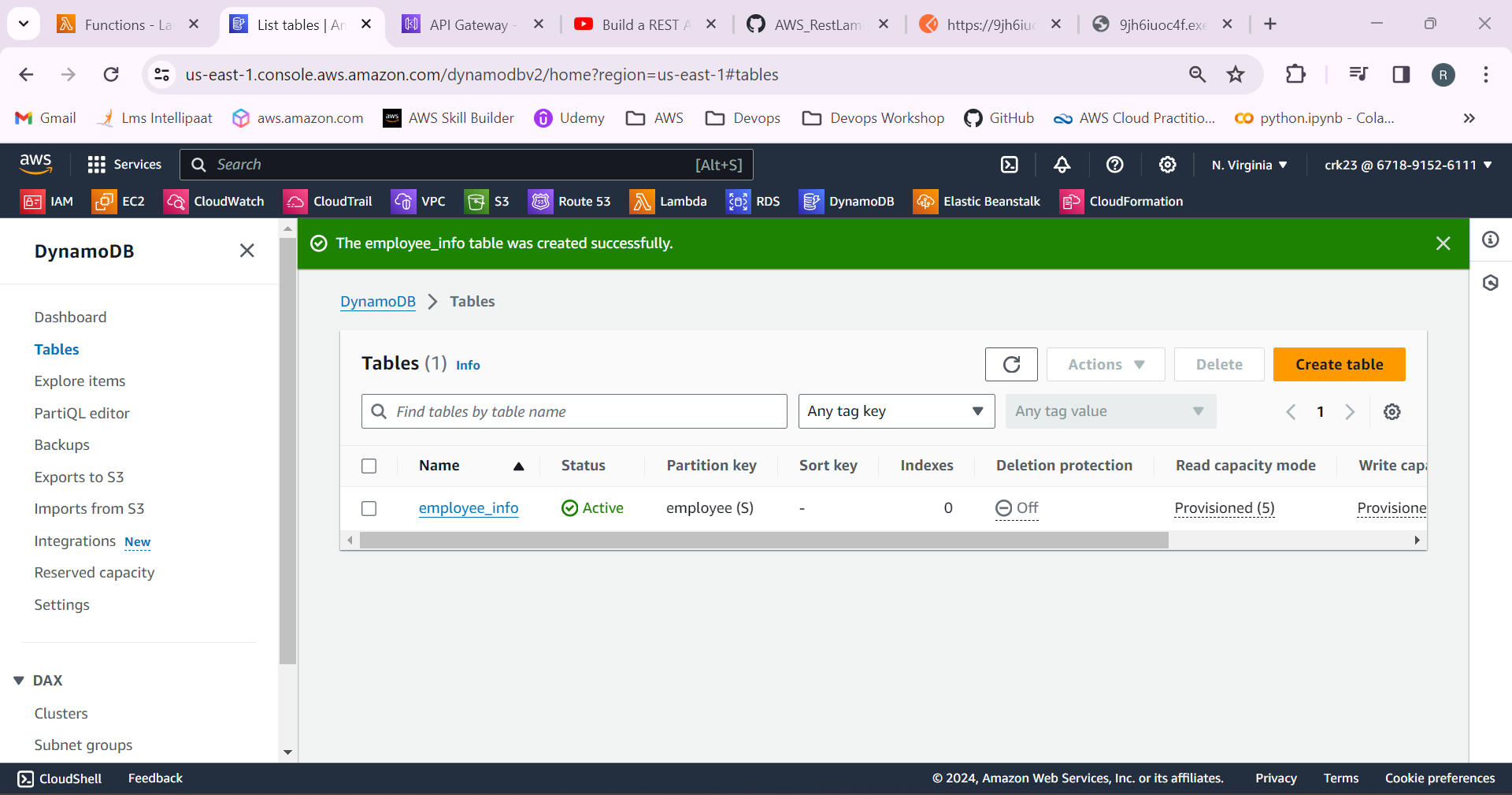
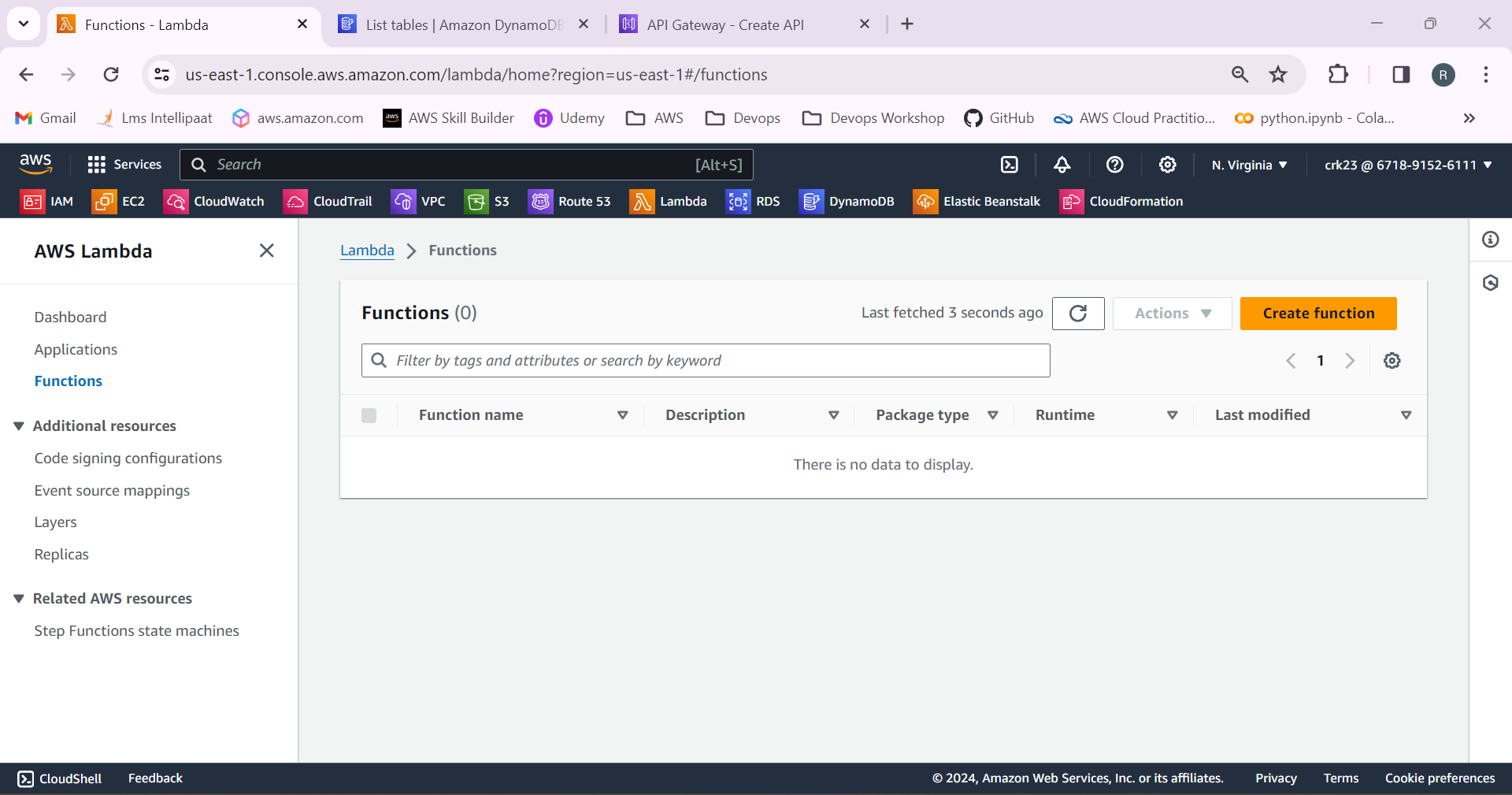
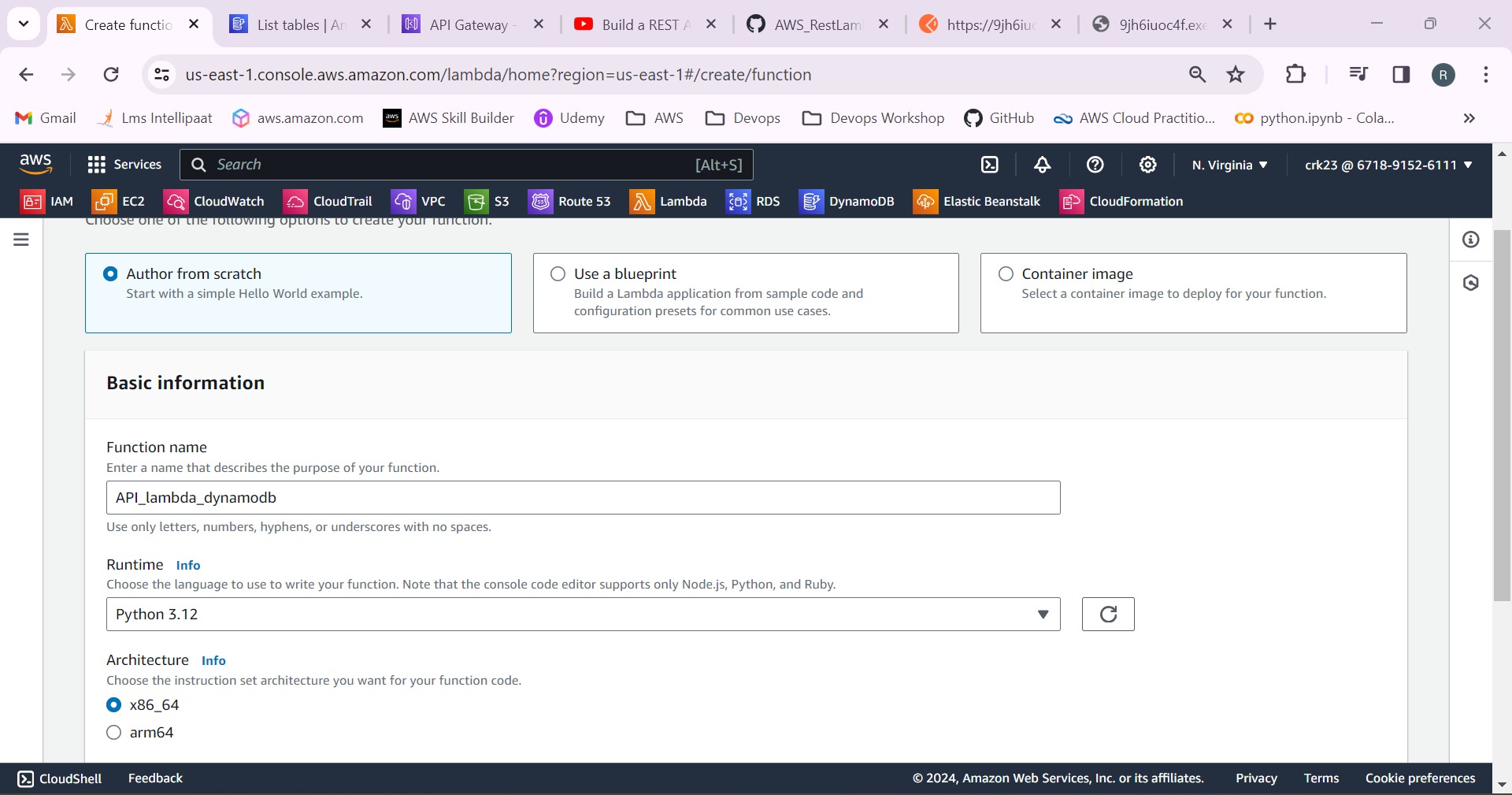


Table has been created.

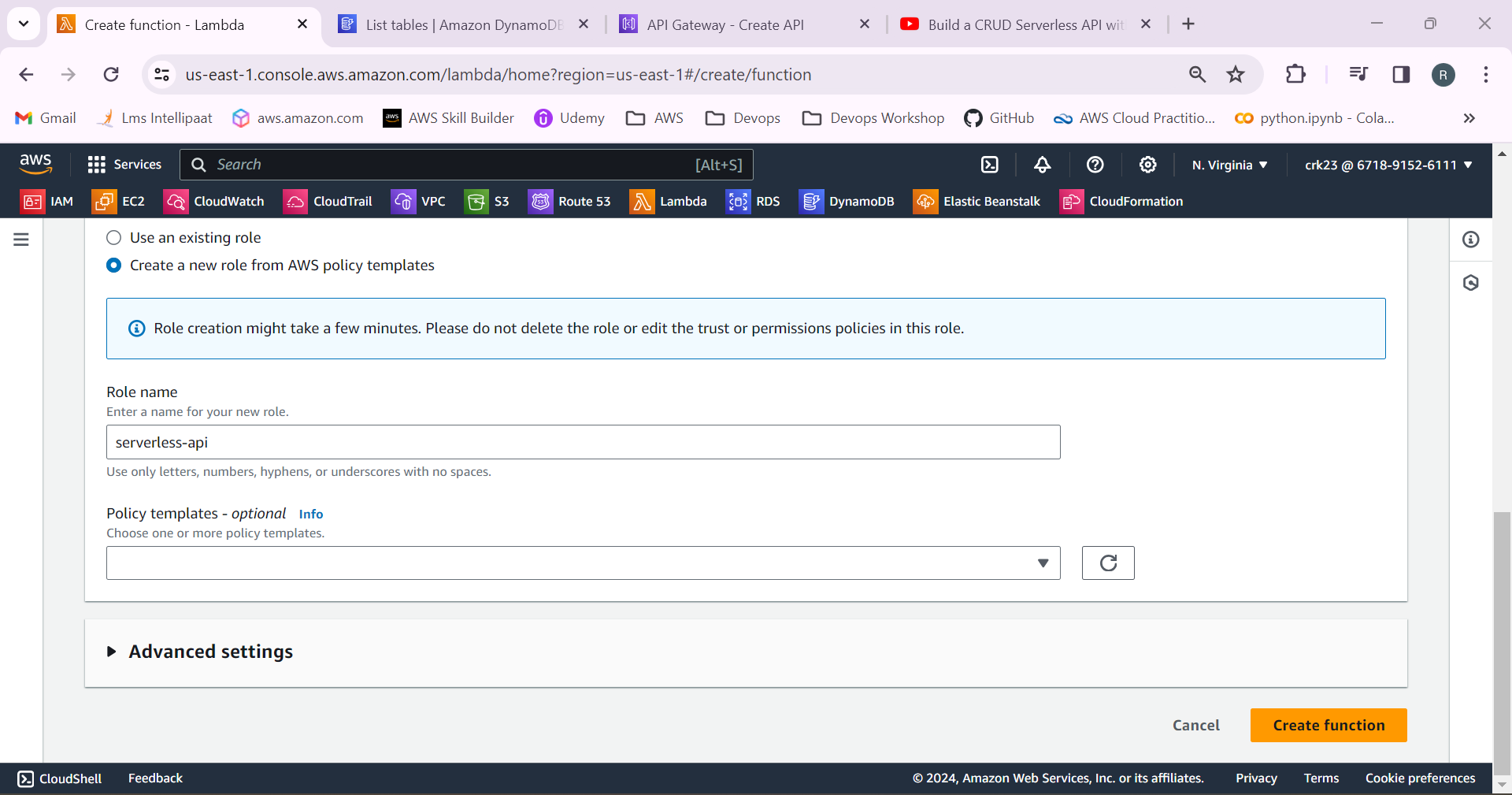


2.Create an lambda function with runtime as python.

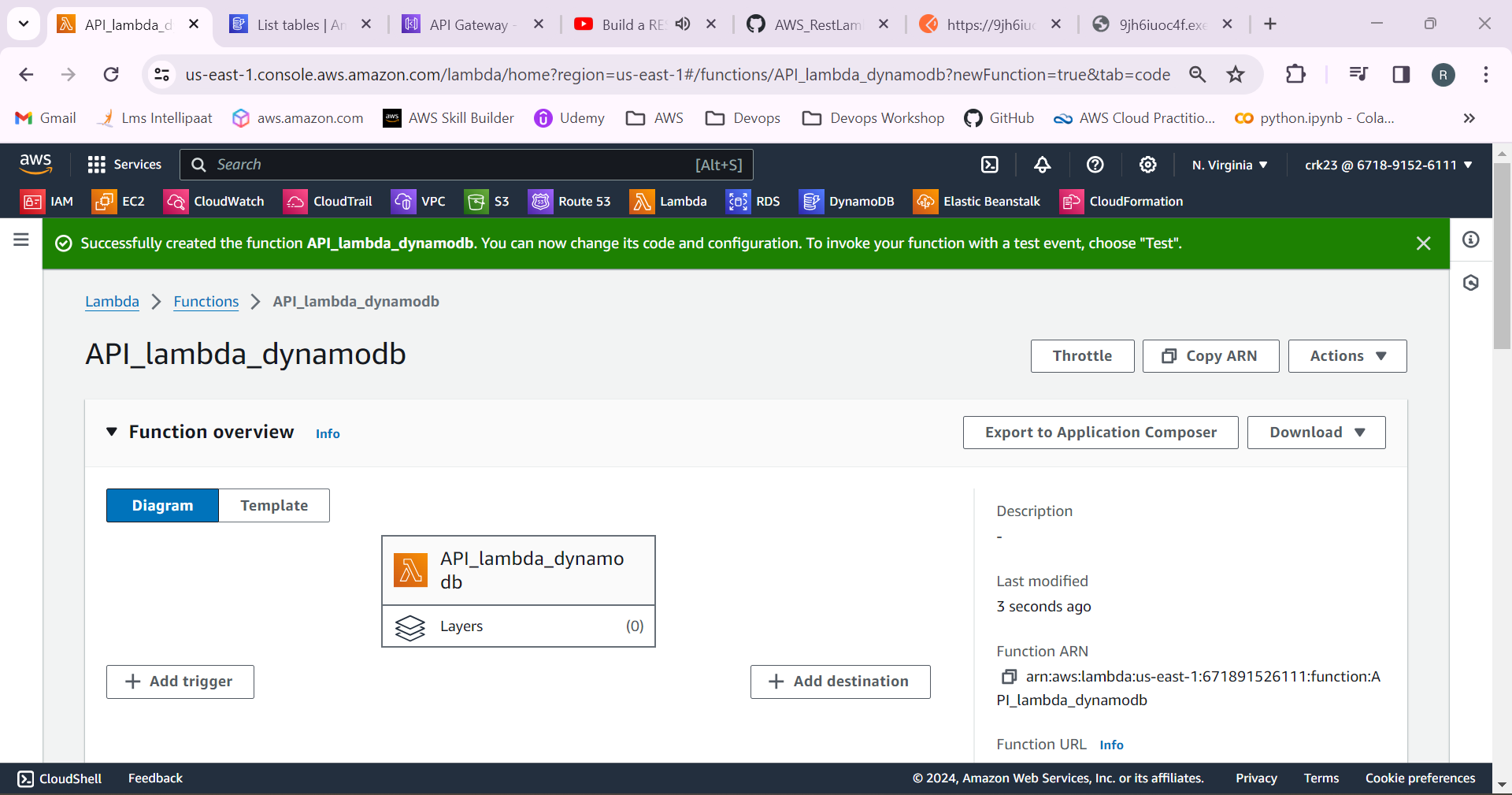




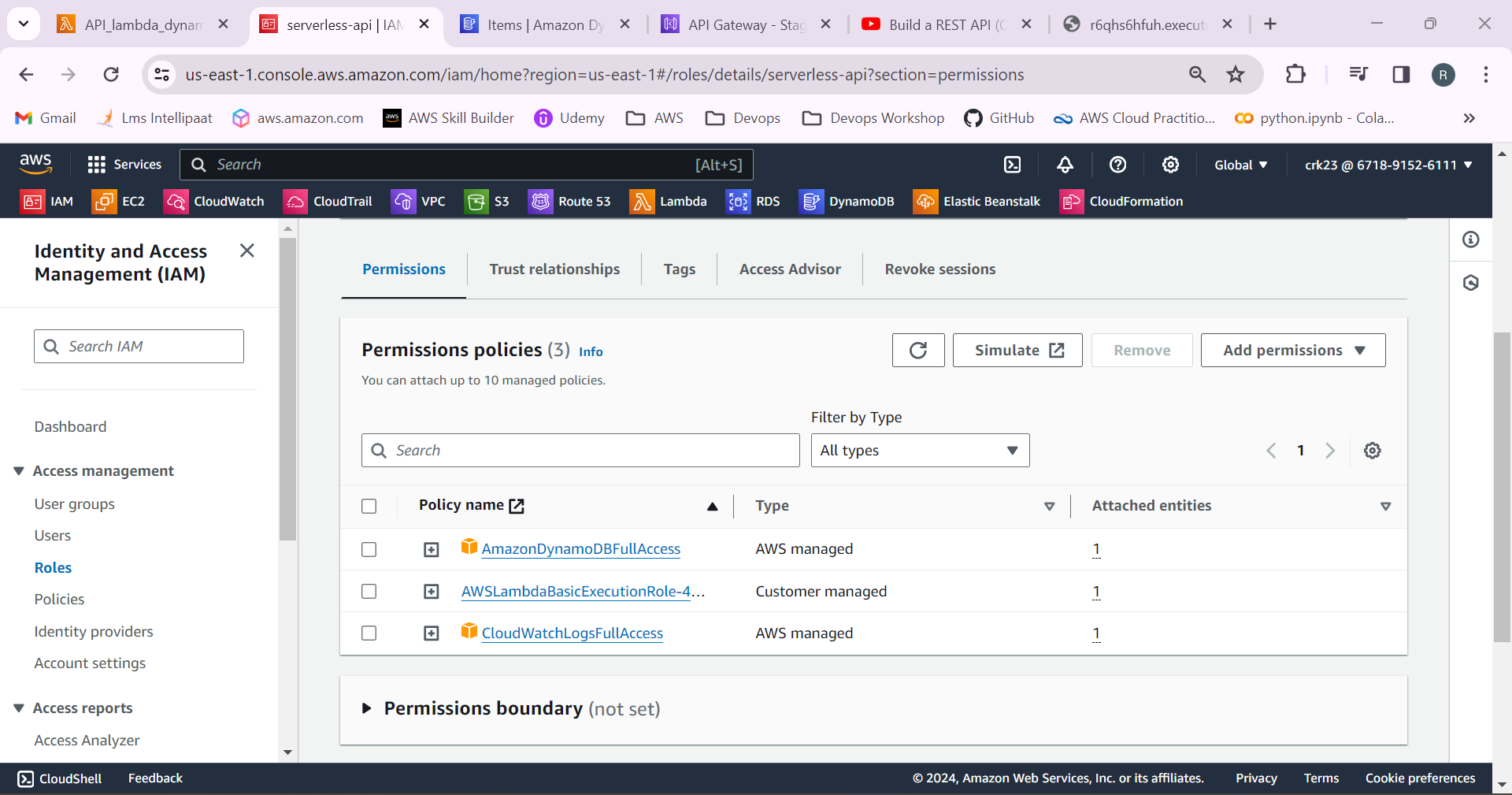
New service role will be created for executing functions.



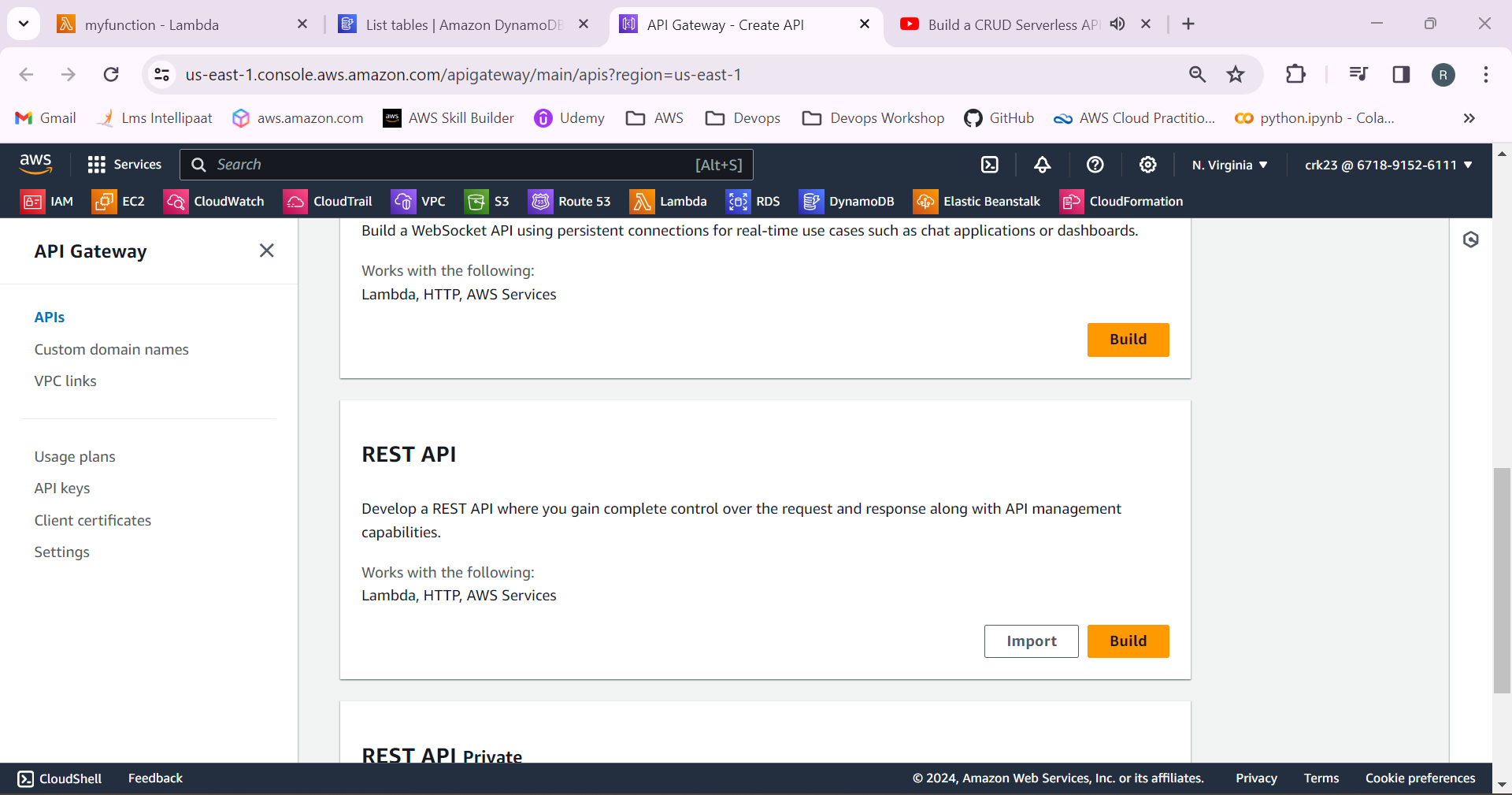
Lambda function has been created.

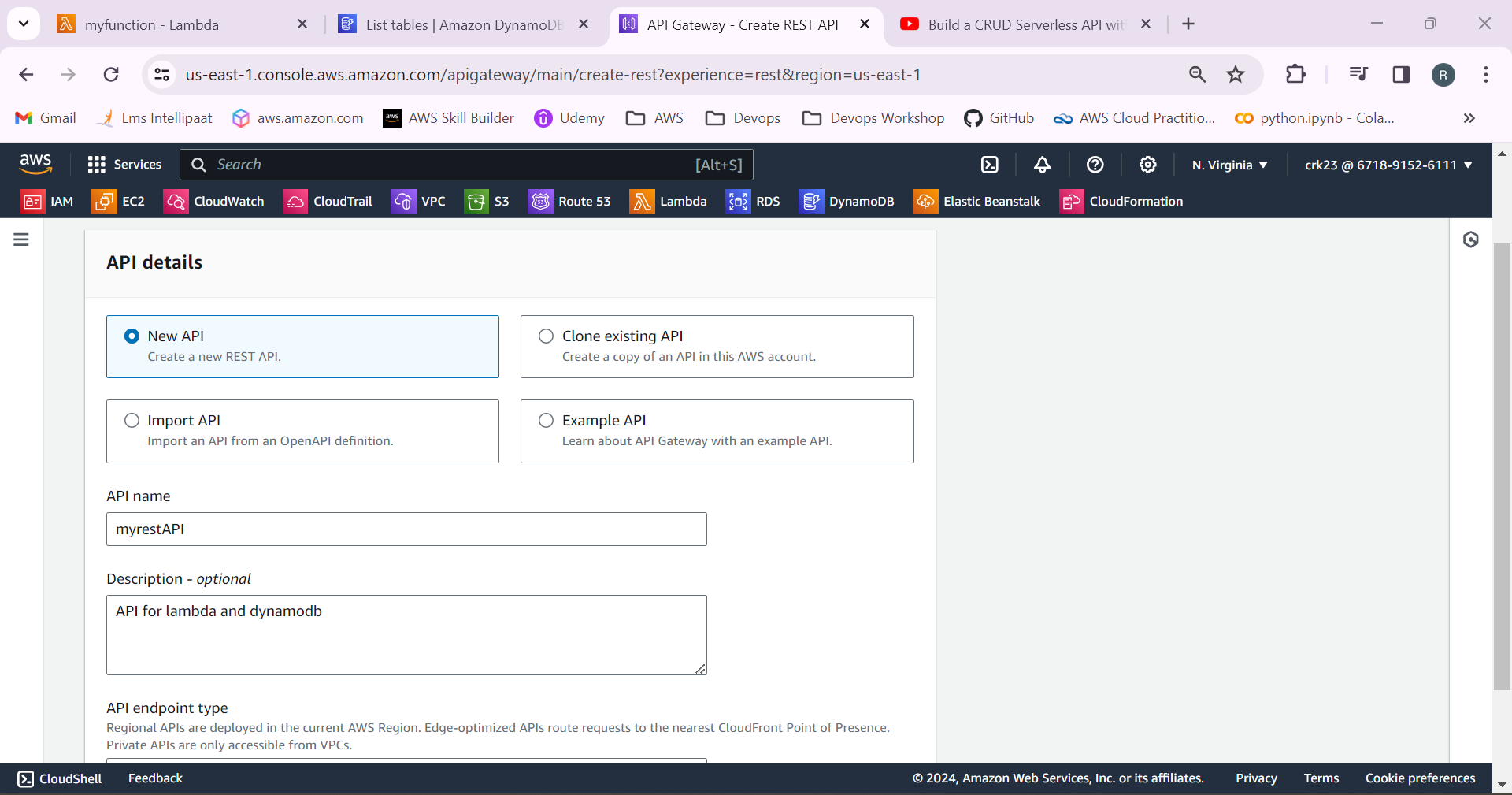


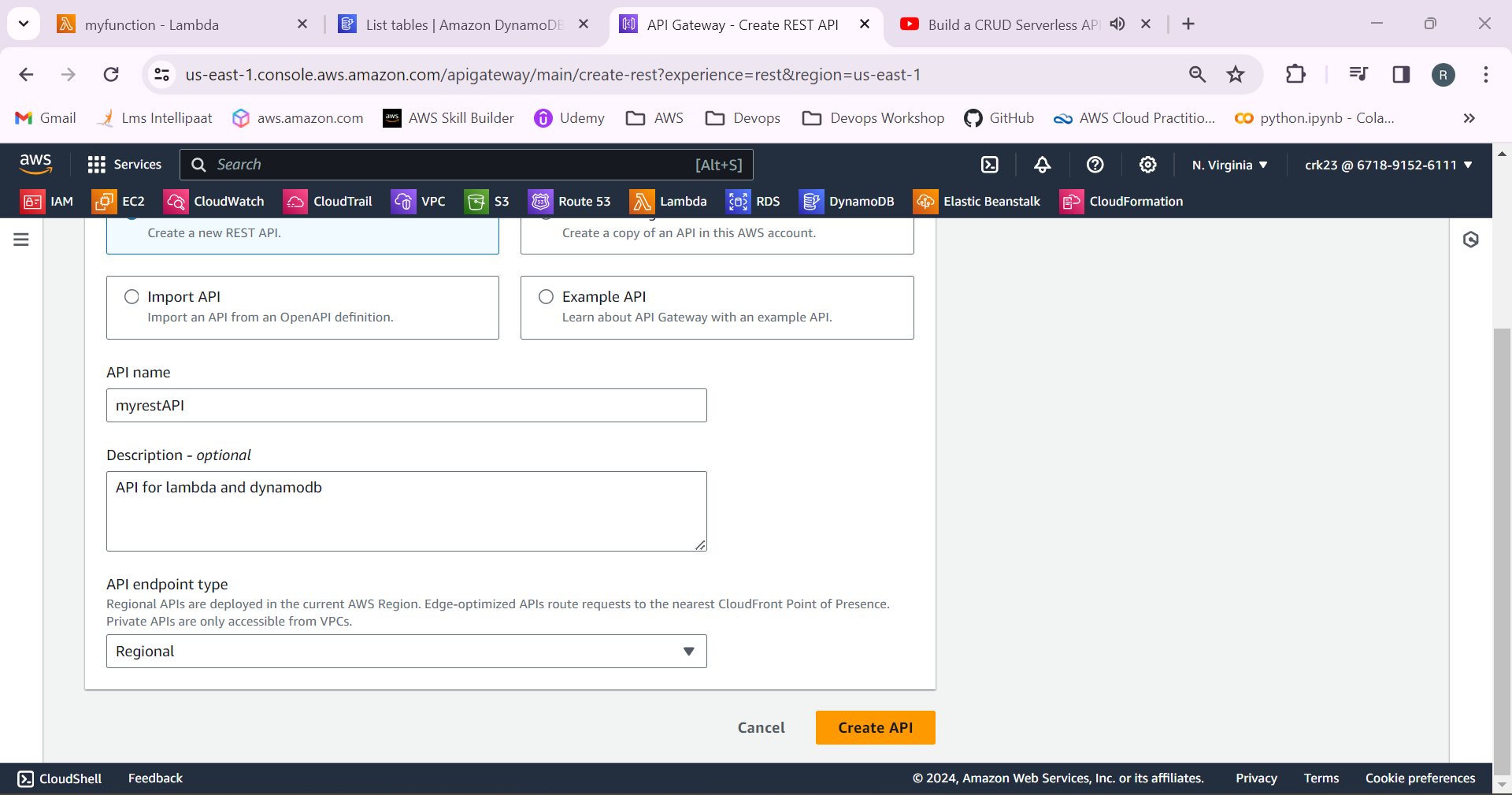
3.Go to IAM > Roles and select the role created from the lambda and add DynamoDB and cloudwatch logs full access permissions to the lambda role.



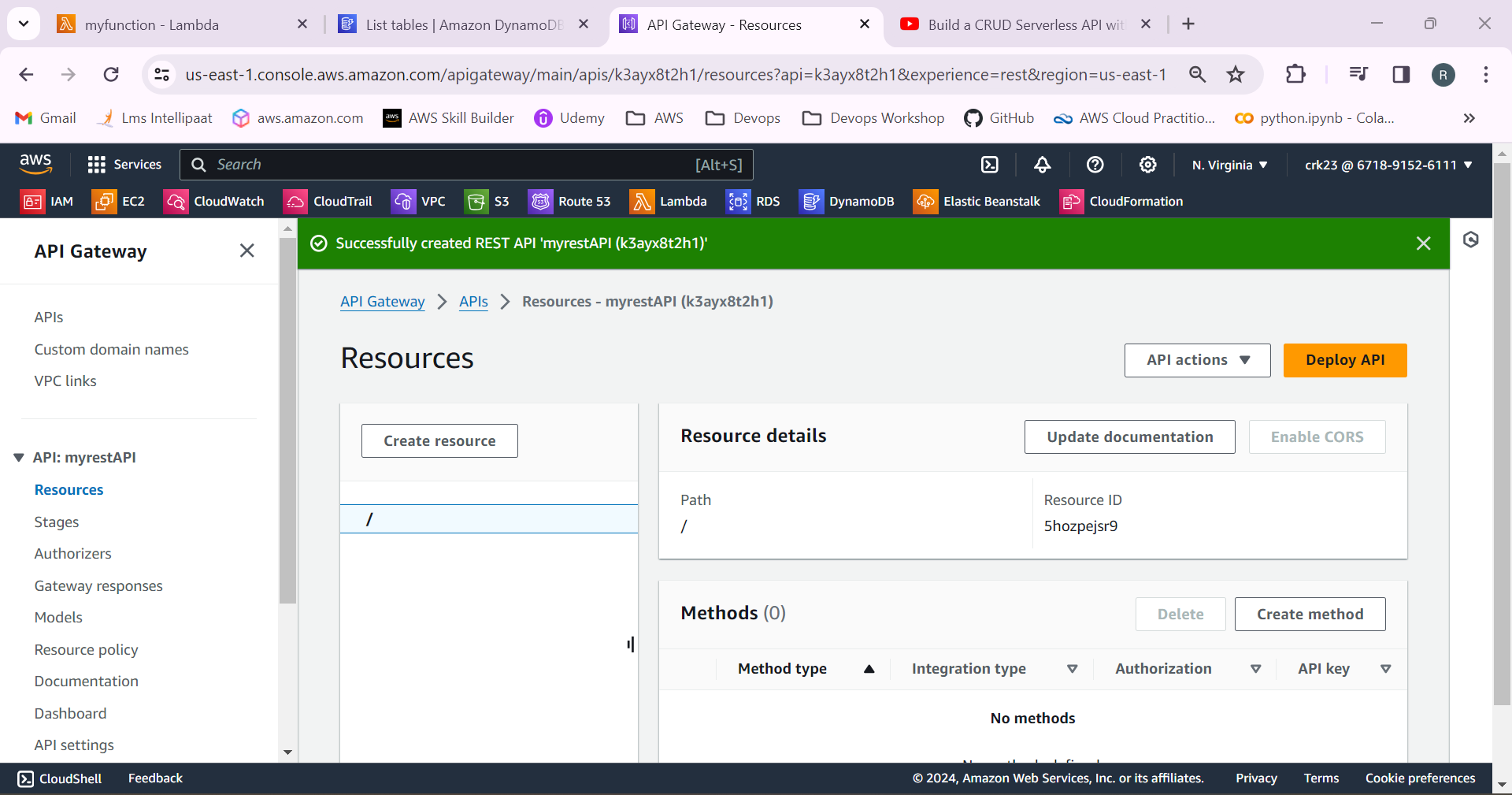
4.Go to API gateway and created a REST API.

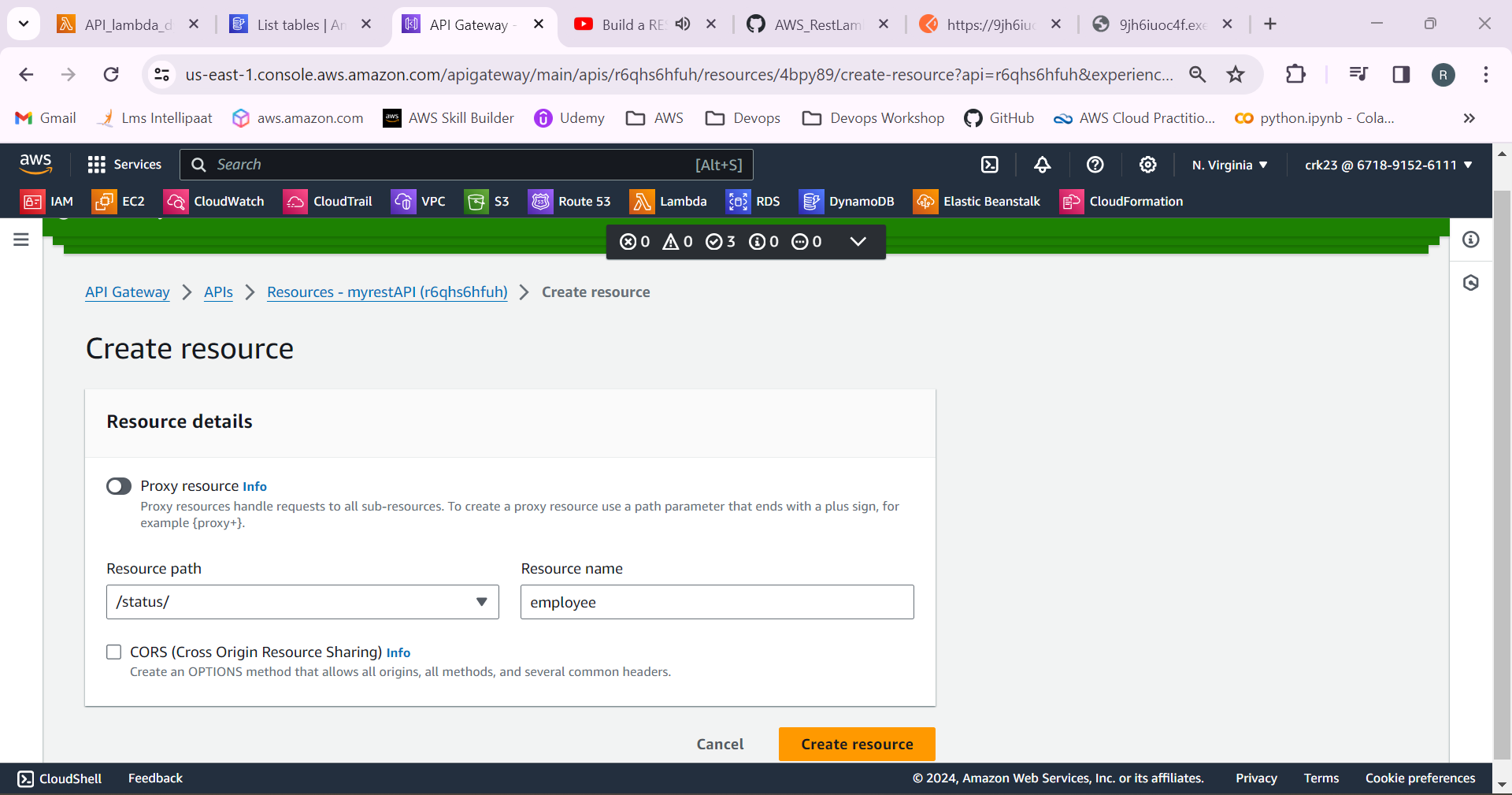


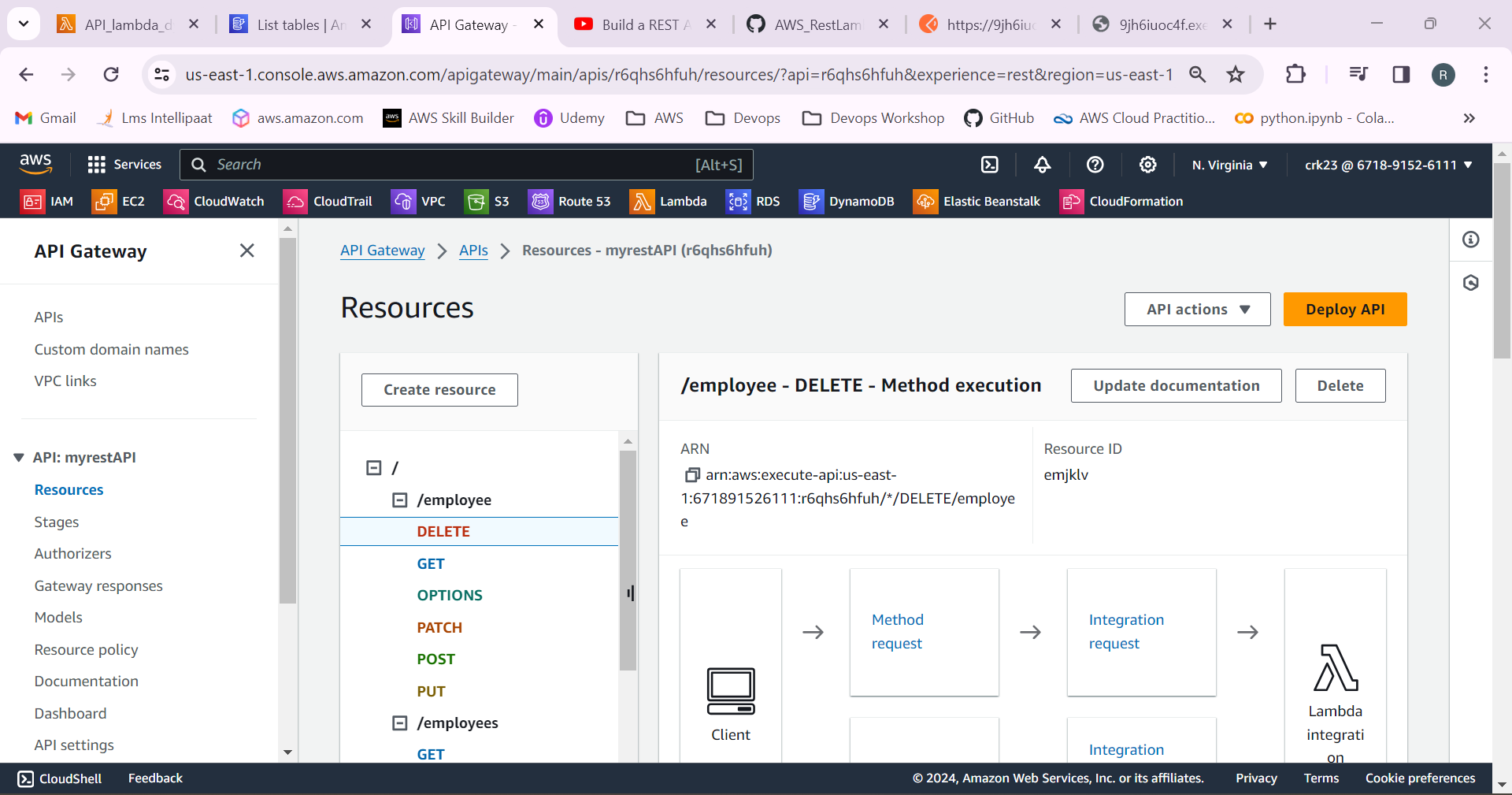




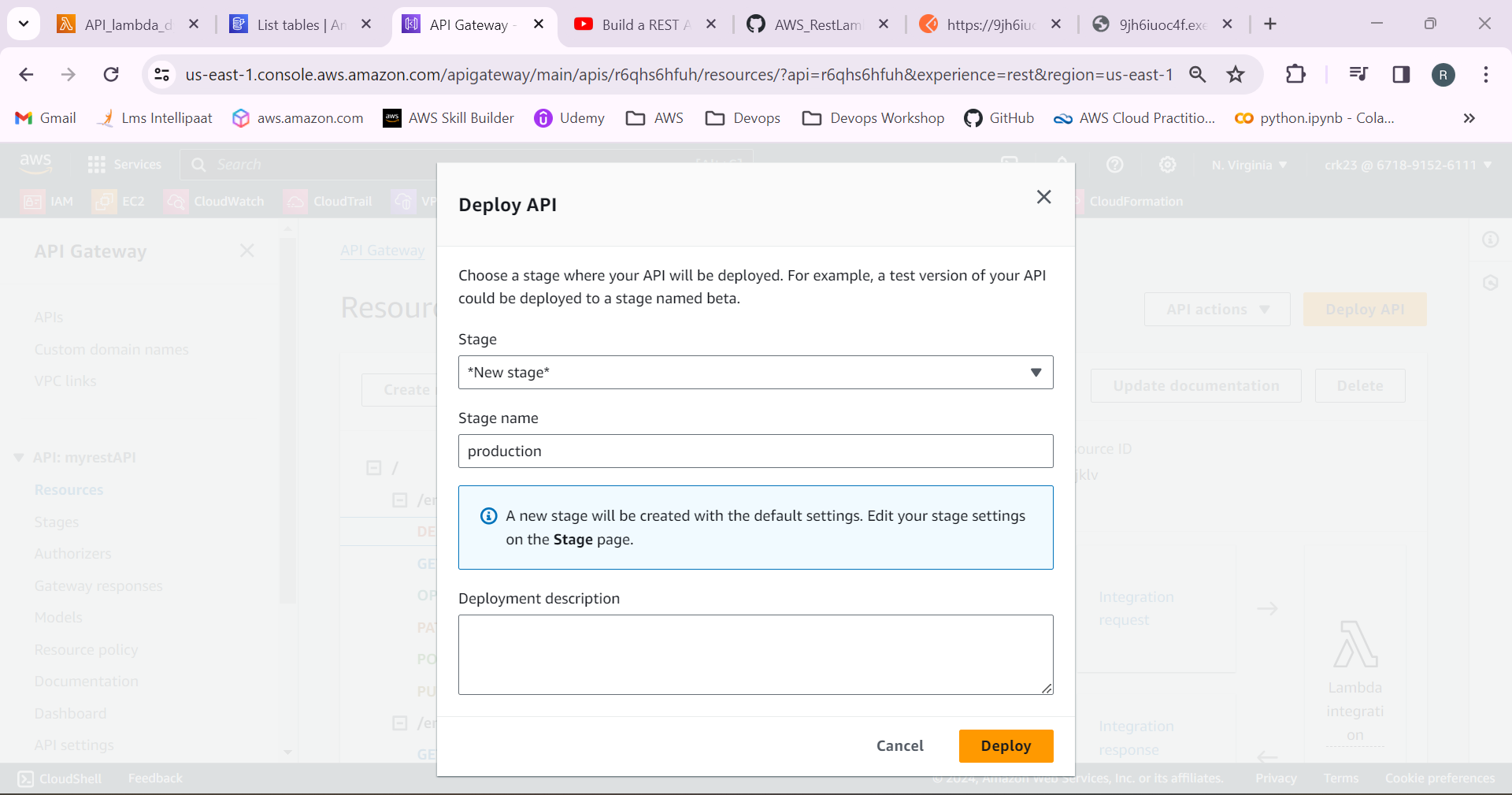
5.Create a resource named employee, employees and status and add GET, PUT, POST, DELETE, OPTIONS methods to the resources.



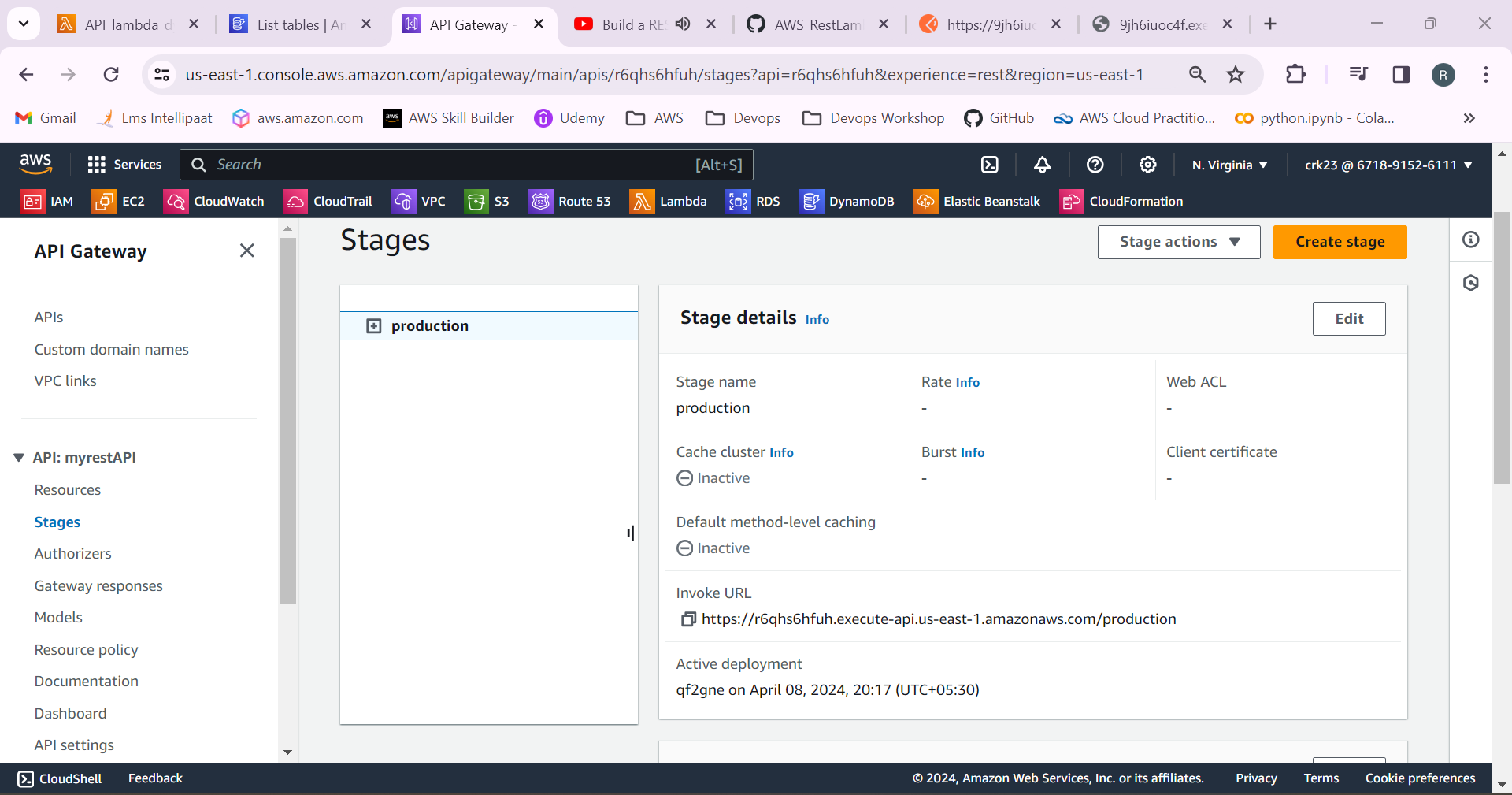




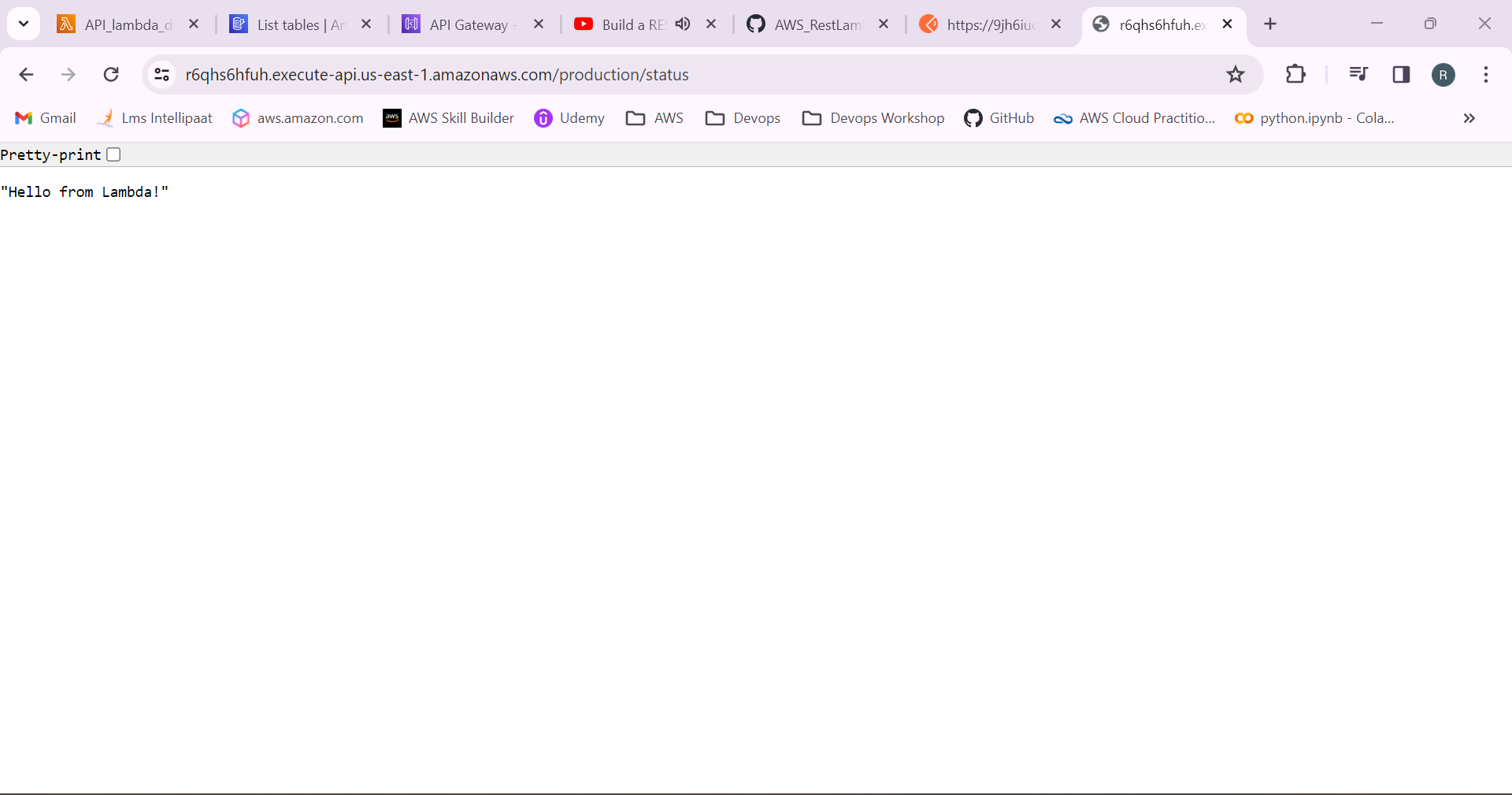
6.Deploy the API.



7.New stage will be created and URL will be generated.

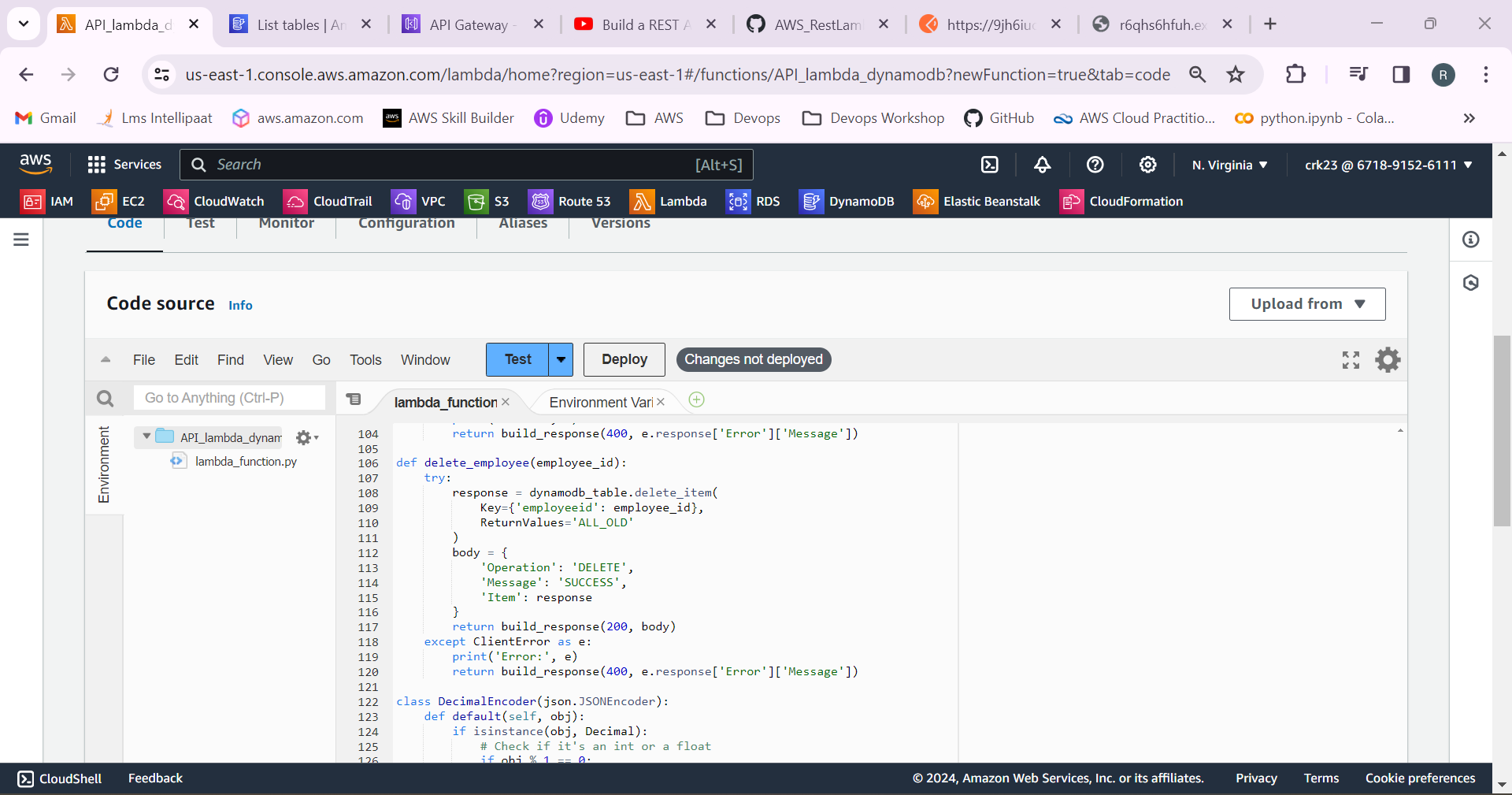


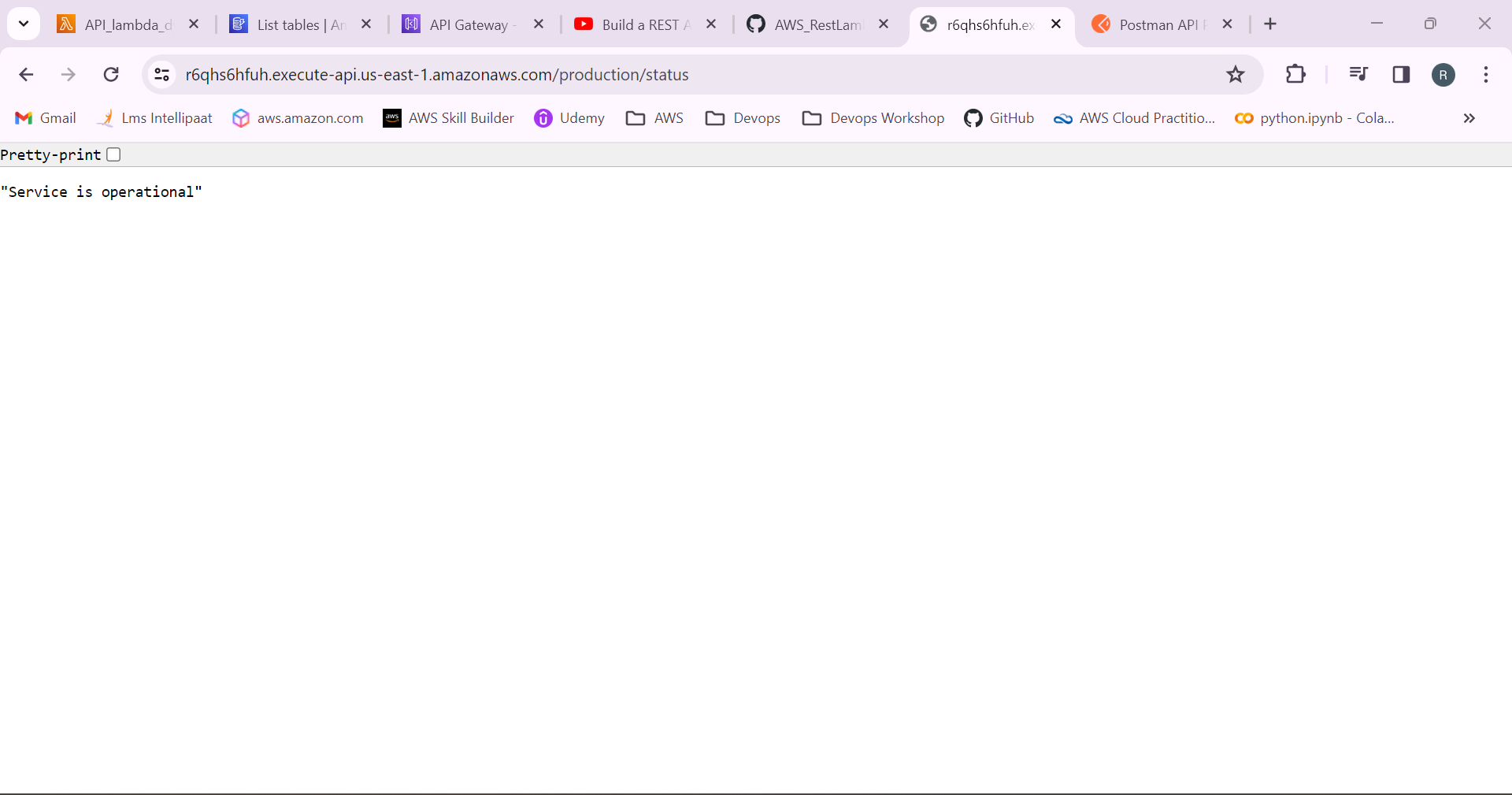
8.Paste the URL in the browser with URL/status to see its working.



9.Go to the lambda function and add the python code to implement CRUD operations.

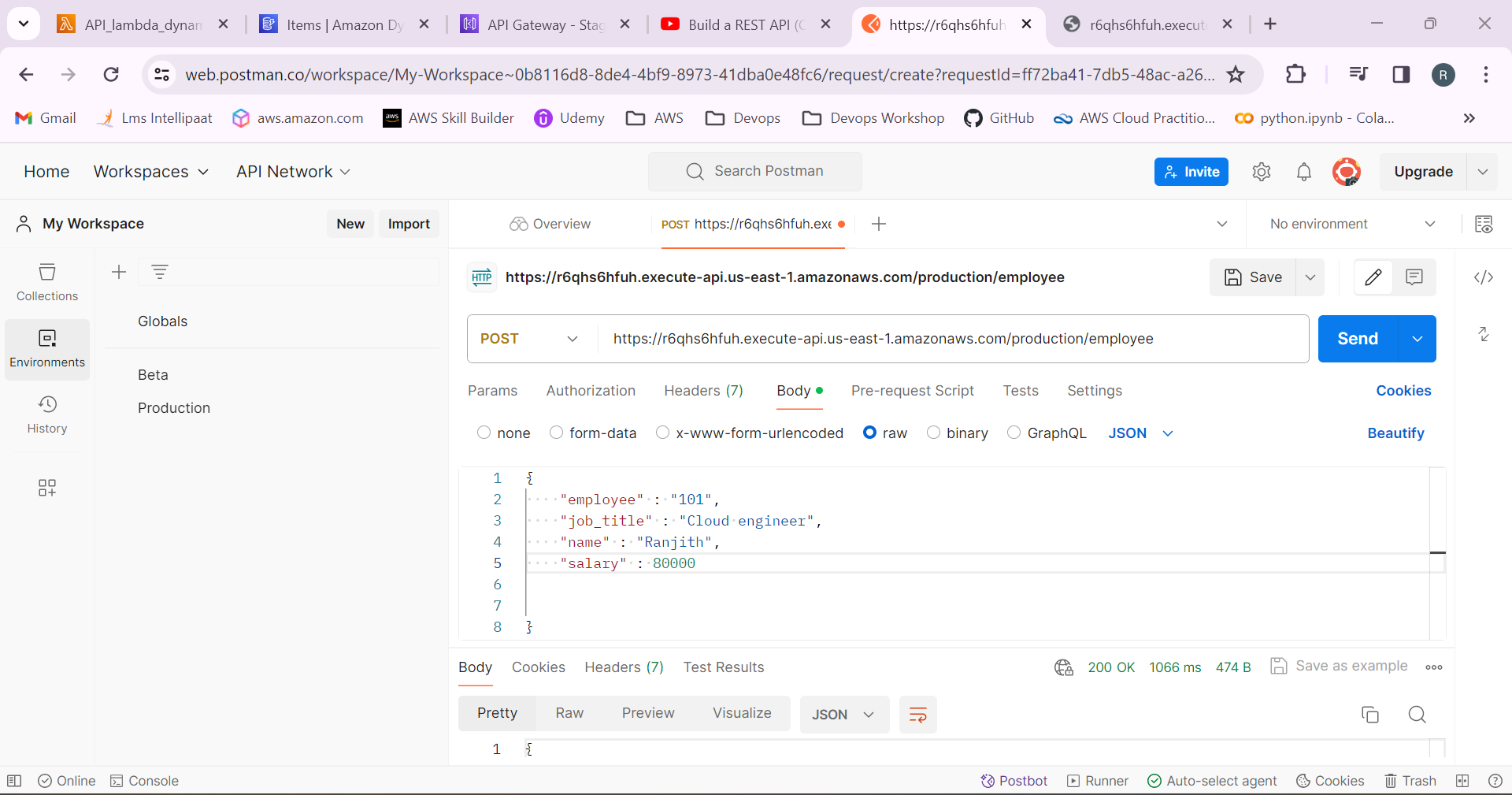
Test and Deploy the code.

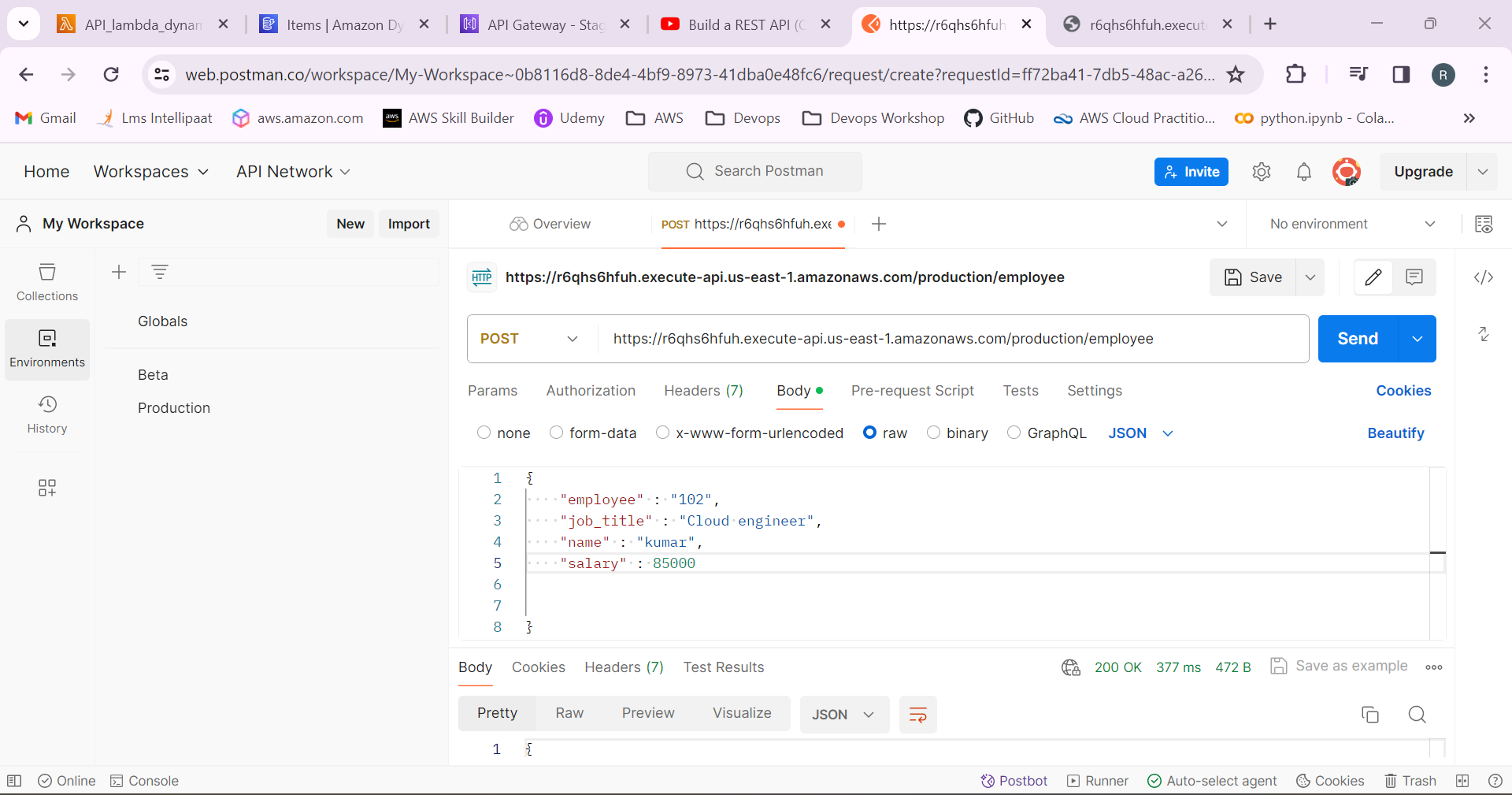




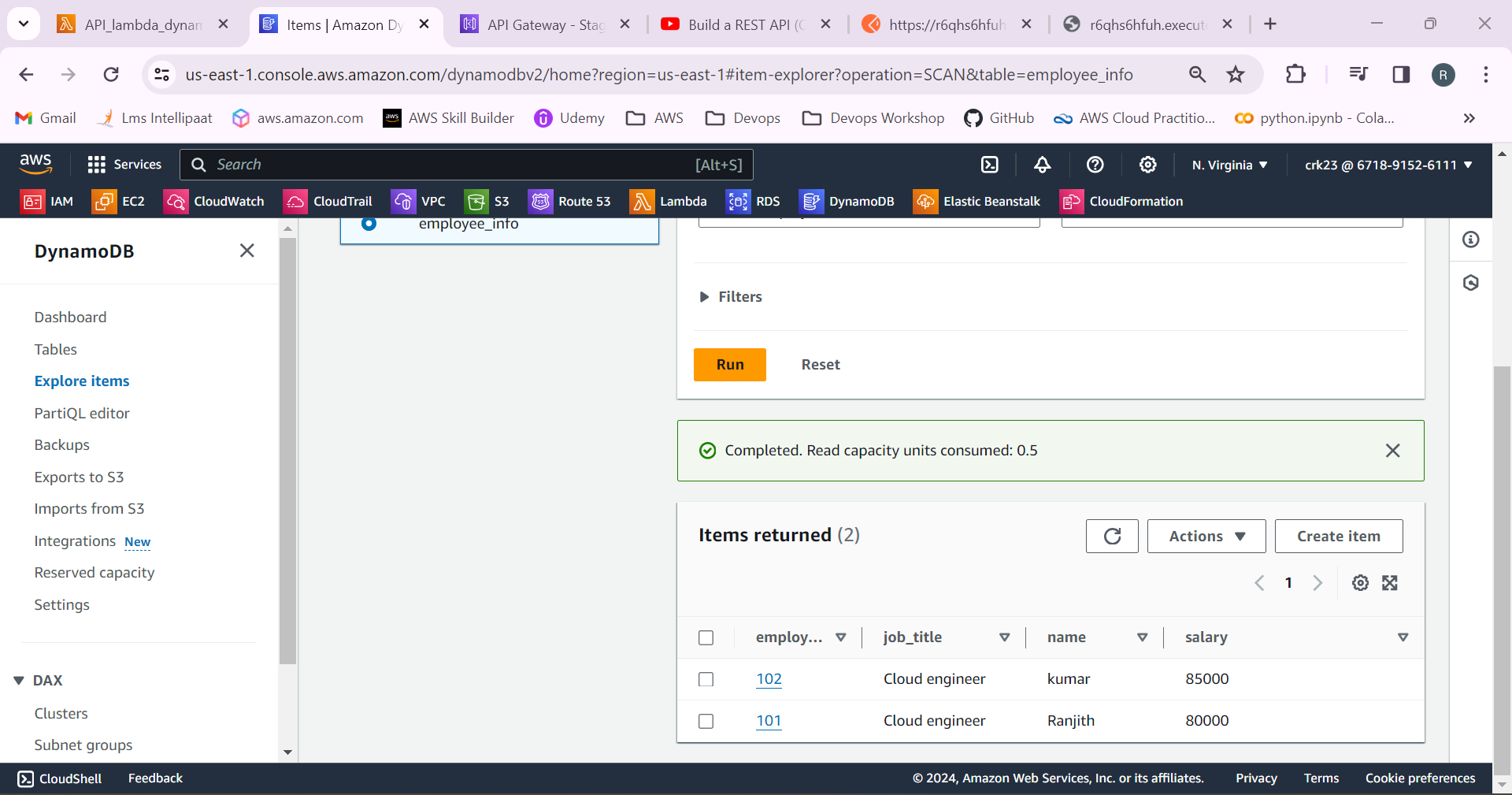
10.Open the Postman and paste the URL and select the operations to reflect in the dynamoDB table.

After entering the values click Send.



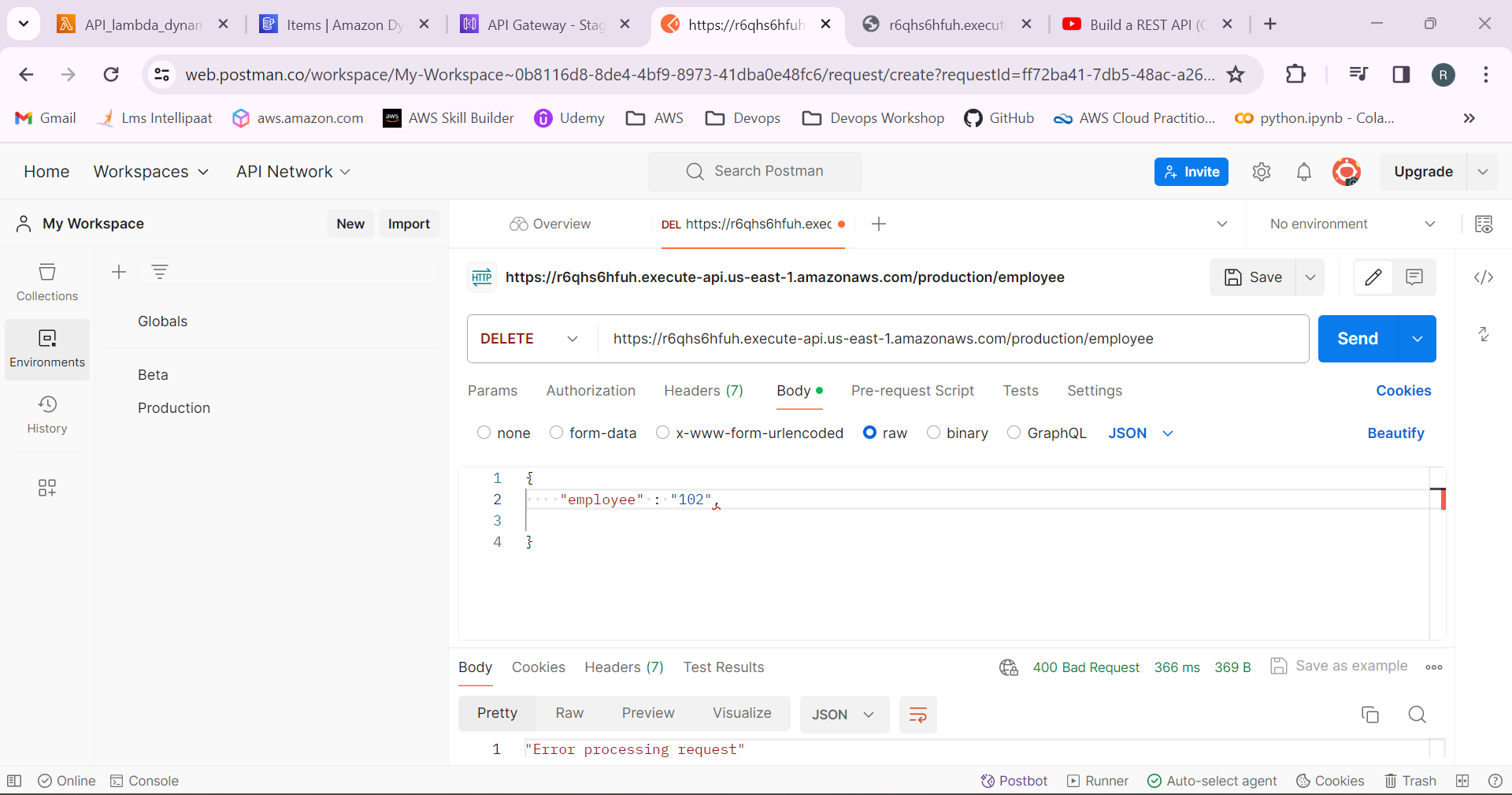


11.Now go to the DynamoDB table and check whether the details entered has been added.



Data has been added successfully!

12.likewise, We can perform all other CRUD operations.



13.Go to cloudwatch and check the logs.

